

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

Patent Application No. 09/418,943

Applicant: Timothy Charles Sowell

Filed: October 15, 1999

TC/AU: 3692

Examiner: Nguyen, Nga B.

Docket No.: 202231

Customer No.: 23460

APPELLANT'S REPLY UNDER 37 C.F.R. SECTION 41.41

Mail Stop Appeal Brief – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

This paper is filed in response to the Examiner's Answer mailed on **August 10, 2007**.
Appellant hereby respectfully requests allowance of the pending claims for the reasons set forth
in Appellant's Brief filed on **December 7, 2004**, and for the further reasons stated herein.

Status of Claims

Pending claims 1-12, 14-44, 53-55 and 74-77 stand rejected, and the rejection of these claims is presently being appealed.

Claims 45-52 and 56-73 were previously withdrawn from consideration in response to a restriction requirement. Claim 13 was previously deleted.

For convenience, a complete listing of the pending claims appears in the Claims Appendix.

Grounds of Rejection to be reviewed on Appeal

The grounds of rejection to be reviewed on appeal are the grounds stated in the Office Action mailed on July 7, 2004. In particular, Appellant appeals:

1. The rejection of claims 1-3, 5-7, 9-12, 14-25, 27-35, 37-40, 42-44, 53-55 and 74-77 as unpatentable under 35 U.S.C. §103(a) as obvious over Archibald et al. U.S. Pat. No. 5,825,883 (Archibald) in view of Knapton, III U.S. Pat. No. 6,363,486 (Knapton) and Sobeski, U.S. Pat. No. 6,499,035 (Sobeski).
2. The rejection of claims 4, 8, 26, 36, and 41 as unpatentable under 35 U.S.C. §103(a) as obvious over Archibald in view of Knapton, Sobeski, and Ahmad, U.S. Pat. No. 5,925,127 (Ahmad).

Argument in Reply to the Answer

Appellant has carefully reviewed the Answer. The Grounds of Rejection Section (9) of the Answer, beginning at page 3 and ending at page 14, restates the grounds recited in the Office Action from which the present appeal was taken. Appellant has addressed the Answer's grounds for rejection in the previously filed appeal brief, and thus Appellant will not repeat these arguments. However, the discussion below reiterates the points previously made in responding to the Response to Argument Section (10) of the Answer.

Appellant seeks reversal of the rejection of the presently pending claims (provided in the Claims Appendix attached hereto). The claimed invention is directed to implementing a use-based billing system for software modules based upon monitoring objects created from the software modules. Under the claimed use-based billing system, software modules are freely distributed to customers' computer systems. The software modules include at least one object class from which object instances are created by the customer systems. Thereafter, software use, for which a customer is charged, is measured according to the object instances created from the at least one object class. Thus, usage of the software modules is gauged by monitoring object instances created from the at least one object class. Basing use of software modules, for purposes of billing a customer, upon monitoring object instances created from the software modules is neither disclosed nor suggested by the combined teachings of the cited prior art references.

The Answer, dated August 10, 2007, concedes that the Archibald reference neither discloses nor suggests a method (or program module) for charging customers for software usage according to *monitored customer use of software modules based upon object instances created from an object class contained within the software modules* previously distributed to a customer system. The Answer seeks to modify the system of Archibald through the teachings of Knapton III and Sobeski to render the claimed invention. However, Knapton III merely identifies types of object-oriented programming languages. The Sobeski reference, in contrast to the claimed *monitoring/accounting* function, discloses a license manager that performs a gatekeeping/authorization/anti-pirating function. Neither Knapton nor Sobeski disclose or even remotely suggest monitoring object instances created from previously provided software modules for purposes of charging a customer for use of provided software modules. In view of

the absence of relevant teachings regarding monitoring and charging for the use of software objects, all the currently pending claims should be allowed.

The Combined Teachings of the Prior Art References
Do Not Render the Claimed Invention Obvious

With regard to Group I, the issue to be resolved is whether the combined teachings of the prior art references render the present invention obvious to someone skilled in the art at the time of the invention. For the reasons previously stated in Appellant's Brief, the combined teachings of Archibald, Knapton III and Sobeski do not render the presently pending claims obvious.

Appellant's claimed invention goes beyond merely reciting that customers are charged for using applications that contain objects. Instead, the claims recite that object instances are created from object classes contained within software modules that are distributed to customer systems, and the customers are thereafter *charged in accordance with monitored use of the software modules according to object instances created from object classes within the software modules*.

The Response to Argument section, at pages 14 and 15 of the Answer, does not address the primary shortcoming of the previous grounds for the rejection. Namely, even if one were to combine the teachings of Archibald, Knapton, and Sobeski, they would not render the presently claimed invention. The Answer concedes that Archibald neither discloses nor suggests carrying out an *object* use-based customer charging scheme. The existence of object-oriented programming (Knapton) and regulating/licensing object usage on client systems (Sobeski) at the time Appellant conceived the current invention does not overcome the shortcomings of the Archibald reference teachings and therefore does not support the Answer's obviousness conclusion.

Sobeski does not teach monitoring usage of software modules through created objects. Rather, Sobeski discloses a gatekeeper/policing functionality that prevents usage of objects prior to application of a verification procedure. The teaching of a preclusive/policing functionality in Sobeski, if anything, teaches away from the claimed invention which is based upon registering usage that is thereafter charged to the customer, and, when combined with Archibald and Knapton, suggests a system including a policing function that prevents execution of objects until an associated license has been verified for the object.

Turning to Sobeski's disclosure, Sobeski's teachings do not concern charging a customer based upon monitored use of software modules. Instead, Sobeski discloses regulating/restricting/denying access to software that is not authorized to a user. In particular, Sobeski discloses a license manager that (prospectively) sets a flag 206 (see, Col. 6, lines 13-17) associated with a Java object to ensure that only authorized Java objects are executed on a user's system. Sobeski does not even remotely suggest using the policing function to charge users (retrospectively) based upon monitored use of the Java objects. Therefore, the combined teachings of Sobeski, Knapton III, and Archibald do not, in combination, disclose or suggest the recited invention that is directed to *charging a customer based upon monitored object instances created from software modules* on the customer's computer system.

The remaining portion of Section 10 of the Answer merely repeats, verbatim, grounds for the rejection previously stated in the Grounds of Rejection Section (9) of the Answer, beginning at page 3 and ending at page 14. Appellant has addressed the Answer's grounds for rejection in the previously filed appeal brief, and thus Appellant will not repeat these arguments here.

Conclusion

In summary, the present invention is not rendered obvious from the combined teachings of Archibald, Knapton, Sobeski, and Ahmad. Appellant's claimed method for distributing and charging for use of software modules – as indicated by object instances created from the software modules, is not suggested by the combined teachings of the cited prior art. If anything, the combined teachings of Archibald and Sobeski suggest a pre-authorization function for software modules rather than an arrangement wherein software use is later charged based upon monitored object instances. For these reasons, as well as others stated herein above and in the Appeal Brief, the presently pending claims are patentable over the prior art presently known to Appellant.

Appellant therefore requests reversal of the rejection of the presently pending claims.

Respectfully submitted,



Mark Joy, Reg. No. 35,362
LEYDIG, VOIT & MAYER, LTD.
Two Prudential Plaza
180 North Stetson Ave., Suite 4900
Chicago, Illinois 60601-6731
(312) 616-5600 (telephone)
(312) 616-5700 (facsimile)

Date: October 10, 2007

Claims Appendix

The Pending Appealed Claims Consist of: 1-12, 14-44, 53-55 and 74-77.

1. A method for charging customers for use of software comprising the steps of:
establishing a use-based pricing scheme for a set of software modules;
distributing the set of software modules to a customer, wherein the set of software modules comprise at least one object class from which objects are instantiated on a customer system;
monitoring customer use of the software modules; and
charging the customer according to use of the distributed software modules as determined during the monitoring step, wherein software usage is measured according to object instances created from the at least one object class.
2. The method of claim 1 wherein the customer creates a number of instances from a software module, and use of the software module is measured according to instances detected at a site of the customer during the monitoring step.
3. The method of claim 2 wherein instances created from a software module are periodically accessed to determine use during the monitoring step.
4. The method of claim 3 wherein the monitoring step comprises registering each day that an instance created from a software module is active; and wherein the charging step comprises charging the customer a daily rate for use of the software module.
5. The method of claim 1 further comprising the step of providing a demonstration mode for instances such that instances in the demonstration mode are executable at a customer site without charge.
6. The method of claim 1 further comprising maintaining a single agreement governing use of instances created from the set of software modules for an enterprise.

7. The method of claim 1 further comprising the step of monitoring a termination date for instances derived from a software module having a time-limited duration.

8. The method of claim 7 further comprising the step of issuing a warning in response to detecting an upcoming expiration date for an instance of a software module.

9. The method of claim 1 further comprising the step of maintaining an account for storing units of credit for a customer; and wherein said charging step comprises decrementing the customer's credit account by an appropriate number of units of credit based upon said monitoring step.

10. The method of claim 1 further comprising the step of generating a report summarizing use of software modules at the customer site.

11. The method of claim 1 wherein the charging step is based upon registered uses of a software module.

12. The method of claim 11 wherein the registered uses of a software module are measured according to execution of an instance created from the software module.

Claim 13 was previously deleted.

14. The method of claim 11 wherein the software module is an object class for creating an application engine object.

15. The method of claim 11 wherein the software module is an object class for creating a view engine object.

16. The method of claim 1 wherein the monitoring step comprises determining a time duration that an object instantiated from a software module is active.

17. The method of claim 1 wherein the monitoring step comprises registering execution of an instance that tracks throughput of a process.

18. The method of claim 1 wherein individual ones of the set of software modules are individually priced.

19. The method of claim 1 wherein the set of software modules includes at least a first software module supplied by a third party vendor and further comprising the step of:
compensating a third party vendor based upon a use by a customer of the first software module determined during the monitoring step.

20. The method of claim 1 wherein the distributing step comprises transmitting the set of software modules via a network connection.

21. The method of claim 20 wherein the network connection comprises an Internet connection.

22. The method of claim 1 comprising a step of reporting usage information to a software brokerage facility.

23. The method of claim 22 wherein the reporting step includes identifying the location of an instance created from a software module.

24. The method of claim 1 comprising the step of determining that a license manager has not reported to a software brokerage facility and in response registering a communication failure at a central licensing facility.

25. The method of claim 1 wherein the monitoring step includes storing use information in summary format in a database.

26. The method of claim 1 further comprising the step of issuing a re-ordering reminder to a customer.

27. The method of claim 1 wherein the software modules relate to industrial manufacturing automation software.

28. The method of claim 1 wherein the software modules relate to industrial manufacturing information software.

29. The method of claim 1 further comprising maintaining an agreement governing use of instances created from the set of software modules for an enterprise wherein the instances comprise both lifetime billed and use-based billed instances.

30. The method of claim 1 further comprising the step of providing configuration tools enabling a user to create customized instances from the software modules.

31. A method for vending software in the form of software modules via electronic commerce channels comprising the steps of:

maintaining an electronic commerce site including a software module selection interface, the software module selection interface enabling a customer to request a software module for use at a customer site, wherein the software module comprises at least one object class from which objects are instantiated on a customer system;

providing a software module management framework to the customer for installation at a customer site, wherein the management framework includes components for registering use of the software module at the customer site; and

charging the customer based upon registered use of the software module, wherein software module usage is measured according to object instances created from the at least one object class.

32. The method of claim 31 wherein the use of the software module comprises executing an instance created from the software module.

33. The method of claim 31 wherein the use of the software module comprises creating an instance from the software module.

34. The method of claim 31 wherein registering use of the software module provides a measure of throughput of an industrial process.

35. The method of claim 31 wherein the module management framework supports creation of instances from software modules at the customer cite having differing use modes including at least: a lifetime mode and a use-based mode, and wherein said method comprises the further step of registering execution of instances operating in the use-based mode.

36. The method of claim 35 wherein the use-based mode is measured in days and wherein an instance operating in use-based mode is registered each day in which the instance is executed.

37. A method for charging customers for use of software comprising the steps of:
providing a set of individually identifiable units of software comprising at least one
object class from which objects are instantiated on a customer system;
individually pricing ones of the set of individually identifiable units of software;
authorizing use of the executable software; and
charging a customer based upon use of selected ones of the set of individually identifiable
units of software, and wherein software usage is measured according to object instances created
from the at least one object class.

38. The method of claim 37 wherein the authorizing step comprises transmitting a
license file containing code enabling use by the customer of the executable software.

39. The method of claim 37 further comprising the step of:
integrating self-monitoring process software within the executable software; and
registering use of the executable software by the self-monitoring process.

40. The method of claim 37 wherein the executable software is industrial automation
software.

41. The method of claim 37 wherein the self-monitoring process software comprises
functions for informing the customer of a need to reorder credit to continue using the executable
software.

42. A method for charging customers for use of software comprising the steps of:
first providing a set of software modules for software customers, wherein the set of software modules comprise at least one object class from which objects are instantiated on a customer system;

second providing a software licensing facility including a brokering facility through which software customers pay for software execution units, and wherein the brokering facility includes a set of software customer accounts; and

charging a software customer account a number of software execution value units based upon the value of software modules utilized by a customer, and wherein software module usage is measured according to object instances created from the at least one object class.

43. The method of claim 42 wherein the charging step is performed by an automated billing process.

44. The method of claim 42 comprising the further step of providing an on-line customer interface; and wherein the first providing step includes the step of providing a network interface enabling users to download software modules from a remote location.

Claims 45-52 were previously withdrawn from consideration in view of a restriction requirement.

53. A memory containing a software module structure facilitating automated distribution of software to customers, the software module structure comprising:
a supplier identification;
a product description;
a billing definition; and
an executable program module including at least one object class from which instantiated objects are rendered.

54. The memory containing a software module structure of claim 53 wherein the billing definition includes a usage rate.

55. The memory containing a software module structure of claim 54 wherein the billing definition includes a lifetime rate.

Claims 56-73 were previously withdrawn from consideration in view of a restriction requirement.

74. The method of claim 11 wherein the registered uses of a software module are measured according to creating an object instance from the software module.

75. The method of claim 42 wherein the use of the software module comprises executing an object instance created from the software module.

76. The method of claim 42 wherein the use of the software module comprises creating an object instance from the software module.

77. The method of claim 42 wherein registering use of the software module provides a measure of throughput of an industrial process.